

```

FFF FFFF FFFF FFFF FFFF 111 111 XXX XXX
FFF FFFF FFFF FFFF FFFF 111 111 XXX XXX
FFF FFFF FFFF FFFF FFFF 111 111 XXX XXX
FFF 111111 111111 111111 XXX XXX
FFF 111111 111111 111111 XXX XXX
FFF 111111 111111 111111 XXX XXX
FFF 111 111 111 XXX XXX
FFF 111 111 111 XXX XXX
FFF 111 111 111 XXX XXX
FFF FFFF FFFF FFFF FFFF 111 111 XXX XXX
FFF FFFF FFFF FFFF FFFF 111 111 XXX XXX
FFF FFFF FFFF FFFF FFFF 111 111 XXX XXX
FFF 111 111 111 111 111 111 XXX XXX
FFF 111 111 111 111 111 111 XXX XXX
FFF 111 111 111 111 111 111 XXX XXX
FFF 111 111 111 111 111 111 XXX XXX
FFF 111 111 111 111 111 111 XXX XXX
FFF 1111111111 1111111111 XXX XXX
FFF 1111111111 1111111111 XXX XXX
FFF 1111111111 1111111111 XXX XXX

```

```
IIIIII  NN  NN  IIIII  FFFFFFFF  CCCCCC  PPPPPPP
IIIIII  NN  NN  IIIII  FFFFFFFF  CCCCCC  PPPPPPP
  II    NN  NN  II     FF          CC      PP      PP
  II    NN  NN  II     FF          CC      PP      PP
  II    NNNN NN  II     FF          CC      PP      PP
  II    NNNN NN  II     FF          CC      PP      PP
  II    NN  NN  II     FFFFFFFF  CC      PPPPPPP
  II    NN  NN  II     FFFFFFFF  CC      PPPPPPP
  II    NN  NN  II     FF          CC      PP
  II    NN  NN  II     FF          CC      PP
  II    NN  NN  II     FF          CC      PP
  II    NN  NN  IIIII  FF          CCCCCC  PP
IIIIII  NN  NN  IIIII  FF          CCCCCC  PP
```

```
....
....
....
....
```

```
LL      IIIII  SSSSSSS
LL      IIIII  SSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LL      II     SSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LLLLLLLL IIIII  SSSSSSS
LLLLLLLL IIIII  SSSSSSS
```

```
1 0001 0 MODULE INIFCP (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000',
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine does the one time initialization for FCP.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1 STARLET operating system, including privileged system services
42 0042 1 and internal exec routines. This routine must be called
43 0043 1 in kernel mode.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 16:30
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V03-011 CDS0007 Christian D. Saether 2-May-1984
53 0053 1 Create bad block scanner mailbox as permanent.
54 0054 1
55 0055 1 V03-010 ACG0415 Andrew C. Goldstein, 12-Apr-1984 12:34
56 0056 1 Fix end points of locked area descriptors
57 0057 1
```



V03-009 ACG0408 Andrew C. Goldstein, 24-Mar-1984 0:07  
Misc bug fixes for storage reorganization

V03-008 ACG0408 Andrew C. Goldstein, 23-Mar-1984 12:03  
Dynamically allocate impure storage on startup

V03-007 CDS0006 Christian D. Saether 19-Feb-1984  
Remove reference to INIT\_POOL.  
Change external references to general mode.

V03-006 CDS0005 Christian D. Saether 12-Dec-1983  
Move all GLOBAL data declarations to COMMON.  
Eliminate most of the initialization routine in  
conjunction with the reduction of image sections.  
Get channel by calling IOCSFFCHAN directly instead  
of using SASSIGN (it isn't really assigned to  
a specific device, anyway).

V03-005 CDS0004 Christian D. Saether 27-Aug-1983  
Only assign one disk i/o channel. Remember its ccb address.

V03-004 CDS0003 Christian D. Saether 26-Jul-1983  
Eliminate creation of job controller mailbox.

V03-003 CDS0002 Christian D. Saether 15-Dec-1982  
Remove some non-pic references.

V03-002 CDS0001 C Saether 18-Jul-1982  
Changes to support ACP to XQP file system.

V03-001 LMP0037 L. Mark Pilant, 28-Jun-1982 15:10  
Remove the addressing mode module switch.

V02-004 ACG0245 Andrew C. Goldstein, 23-Dec-1981 21:04  
Add job controller mailbox

V02-003 LMP0004 L. Mark Pilant, 1-Dec-1981 12:10  
Make external references use general mode

V02-002 ACG0167 Andrew C. Goldstein, 16-Apr-1980 19:26  
Previous revision history moved to f11B.REV

LIBRARY 'SYS\$LIBRARY:LIB.L32';  
REQUIRE 'SRC\$FCPDEF.B32';

FORWARD ROUTINE  
INIT\_FCP, ! initialize file system  
INIT\_STORAGE : L\_NORM NOVALUE; ! initialize global storage

! Dummy vectors to bracket the locked down code and data psects.

PSECT GLOBAL = \$AAAAAS (NOWRITE, EXECUTE, ALIGN (9));  
GLOBAL CODE\_START : VECTOR [0];

INIFCP  
V04-000

C 1  
16-Sep-1984 00:37:40  
14-Sep-1984 12:30:32

VAX-11 Bliss-32 V4.0-742  
DISKSVMSMASTER:[F11X.SRC]INIFCP.B32:1 Page 3 (1)

```
: 115      1105 1 PSECT GLOBAL = BLOCKEDC08 (NOWRITE, EXECUTE, ALIGN (9));  
: 116      1106 1 GLOBAL L_CODE_START : VECTOR [0];  
: 117      1107 1  
: 118      1108 1 PSECT GLOBAL = BLOCKEDC98 (NOWRITE, EXECUTE, ALIGN (2));  
: 119      1109 1 GLOBAL L_CODE_END : VECTOR [0];
```

1000  
V04-



```
121 1110 1 GLOBAL ROUTINE INIT_FCP =
122 1111 1
123 1112 1 ++
124 1113 1
125 1114 1 FUNCTIONAL DESCRIPTION:
126 1115 1
127 1116 1 This routine does the one time initialization for FCP.
128 1117 1 It is called during process creation immediately after
129 1118 1 the xcp code is mapped.
130 1119 1
131 1120 1 CALLING SEQUENCE:
132 1121 1 INIT_FCP ()
133 1122 1
134 1123 1 INPUT PARAMETERS:
135 1124 1 NONE
136 1125 1
137 1126 1 IMPLICIT INPUTS:
138 1127 1 system I/O data base
139 1128 1
140 1129 1 OUTPUT PARAMETERS:
141 1130 1 NONE
142 1131 1
143 1132 1 IMPLICIT OUTPUTS:
144 1133 1 IO_CHANNEL: gets channel number of I/O channel
145 1134 1 IO_CCB: gets ccb address of i/o channel
146 1135 1
147 1136 1 ROUTINE VALUE:
148 1137 1 NONE
149 1138 1
150 1139 1 SIDE EFFECTS:
151 1140 1 FCP hooked up to system data base
152 1141 1
153 1142 1 --
154 1143 1
155 1144 2 BEGIN
156 1145 2
157 1146 2 LOCAL
158 1147 2 STORAGE_DESC : VECTOR [2], ! descriptor for allocated storage
159 1148 2 STATUS; ! system status return
160 1149 2
161 1150 2 GLOBAL REGISTER
162 1151 2 BASE = 10;
163 1152 2
164 1153 2 EXTERNAL
165 1154 2 CTL$GL_CTLBASVA : ADDRESSING_MODE (ABSOLUTE),
166 1155 2 ! base address of permanent P1 space
167 1156 2 EXE$GL_FLAGS : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
168 1157 2 ! system flags vector
169 1158 2
170 1159 2 EXTERNAL LITERAL
171 1160 2 STORAGE_SIZE : UNSIGNED (16), ! size of impure area in bytes
172 1161 2 STORAGE_OFFSET : UNSIGNED (16), ! offset to point base register at
173 1162 2 EXE$V_INIT : UNSIGNED (6); ! bit position of FCP init flag
174 1163 2
175 1164 2
176 1165 2 ! First allocate the impure storage region.
177 1166 2
```

```
178 1167
179 1168 P IF NOT (STATUS = $EXPREG (PAGCNT = (STORAGE_SIZE + 511) / 512,
180 1169 P RETADR = STORAGE_DESC,
181 1170 P REGION = 1
182 1171 P ))
183 1172 THEN $EXIT (CODE = .STATUS);
184 1173 BASE = .STORAGE_DESC[1] + STORAGE_OFFSET;
185 1174 CTL$GL_CTLBASVA = .STORAGE_DESC[1];
186 1175
187 1176 INIT_STORAGE ();
188 1177
189 1178 ! Finally set the FCP init'd bit in the system flags word to indicate that
190 1179 ! a file system now exists (significant only during system startup).
191 1180
192 1181 IF TESTBITS (EXESGL_FLAGS [EXESV_INIT])
193 1182 THEN
194 1183
195 1184 ! This will happen when the xgp is merged into the sysinit process.
196 1185 ! It should have all the privileges we need to create this mailbox,
197 1186 ! so elevating and restoring them is not necessary.
198 1187
199 1188
200 1189 BEGIN
201 1190 LOCAL
202 1191 MBX_CHAN,
203 1192 DESC : VECTOR [2];
204 1193
205 1194 PIC_DESC ('ACP$BADBLOCK_MBX', DESC);
206 1195
207 1196 IF NOT $CREMBX (CHAN = MBX_CHAN,
208 1197 P MAXMSG = BB$SC_LENGTH,
209 1198 P BUFOUO = BB$SC_LENGTH*100,
210 1199 P PROMSK = %X'FFFF',
211 1200 P LOGNAM = DESC,
212 1201 P PRMFLG = 1)
213 1202 THEN
214 1203 BUG_CHECK (XQPERR);
215 1204
216 1205 $DASSGN (CHAN = .MBX_CHAN);
217 1206
218 1207 END;
219 1208
220 1209 SSS_NORMAL
221 1210
222 1211
223 1212 ! end of routine INIT_FCP
```

```
.TITLE INIFCP
.IDENT \V04-000\
```

```
.PSECT $CODE$,NOWRT,2
```

```
42 4D 5F 4B 43 4F 4C 42 44 41 42 24 50 43 41 0000 P.AAA: .ASCII \ACP$BADBLOCK_MBX\
58 0000F
```

```
.PSECT $LOCKEDC9$,NOWRT,2
```



; Routine Size: 125 bytes, Routine Base: \$CODE\$ + 0010



INIFCP  
V04-000

6 1  
16-Sep-1984 00:37:40  
14-Sep-1984 12:30:32

VAX-11 Bliss-32 V4.0-742  
DISKSVMSMASTER:[F11X.SRC]INIFCP.B32;1 Page 7 (2)

LOC  
V04-

```
225 1213 1 GLOBAL ROUTINE INIT_STORAGE : L_NORM NOVALUE =
226 1214 1
227 1215 1 ++
228 1216 1
229 1217 1 FUNCTIONAL DESCRIPTION:
230 1218 1
231 1219 1 This routine initializes the file system's global impure area.
232 1220 1
233 1221 1 CALLING SEQUENCE:
234 1222 1 INIT_STORAGE ()
235 1223 1
236 1224 1 INPUT PARAMETERS:
237 1225 1 NONE
238 1226 1
239 1227 1 IMPLICIT INPUTS:
240 1228 1 system I/O data base
241 1229 1
242 1230 1 OUTPUT PARAMETERS:
243 1231 1 NONE
244 1232 1
245 1233 1 IMPLICIT OUTPUTS:
246 1234 1 IO_CHANNEL: gets channel number of I/O channel
247 1235 1 IO_CCB: gets ccb address of i/o channel
248 1236 1
249 1237 1 ROUTINE VALUE:
250 1238 1 NONE
251 1239 1
252 1240 1 SIDE EFFECTS:
253 1241 1 FCP hooked up to system data base
254 1242 1
255 1243 1 --
256 1244 1
257 1245 2 BEGIN
258 1246 2
259 1247 2 LOCAL
260 1248 2 LOCKED_DESC : VECTOR [2], ! descriptor for locked down pages
261 1249 2 STATUS; ! system status return
262 1250 2
263 1251 2 BIND_COMMON;
264 1252 2
265 1253 2 EXTERNAL
266 1254 2 CTL$GL_F11BXQP : ADDRESSING_MODE (ABSOLUTE);
267 1255 2 ! pointer to XQP
268 1256 2
269 1257 2 EXTERNAL LITERAL
270 1258 2 STORAGE_SIZE : UNSIGNED (16), ! size of impure area in bytes
271 1259 2 STORAGE_OFFSET : UNSIGNED (16); ! offset to point base register at
272 1260 2
273 1261 2 LINKAGE
274 1262 2 L_FFCHAN = JSB : GLOBAL (CHANNEL=1, CCB=2);
275 1263 2
276 1264 2 GLOBAL REGISTER
277 1265 2 CHANNEL = 1,
278 1266 2 CCB = 2 : REF BBLOCK;
279 1267 2
280 1268 2 EXTERNAL ROUTINE
281 1269 2 IOC$FFCHAN : L_FFCHAN ADDRESSING_MODE (GENERAL),
```

```
282 1270 DISPATCH;
283 1271
284 1272
285 1273
286 1274 ! Now lock appropriate areas into the working set. These are code and data
287 1275 ! that are used at raised IPL, plus the private kernel stack.
288 1276
289 1277
290 1278 LOCKED_DESC [0] = L_CODE_START;
291 1279 LOCKED_DESC [1] = L_CODE_END - 1;
292 1280
293 1281 STATUS = $LKWSET (INADR = LOCKED_DESC);
294 1282 IF NOT .STATUS THEN $EXIT (CODE = .STATUS);
295 1283
296 1284 LOCKED_DESC [0] = L_DATA_START;
297 1285 LOCKED_DESC [1] = L_DATA_END - 1;
298 1286
299 1287 STATUS = $LKWSET (INADR = LOCKED_DESC);
300 1288 IF NOT .STATUS THEN $EXIT (CODE = .STATUS);
301 1289
302 1290 ! Find an I/O channel for use by the file system.
303 1291
304 1292
305 1293 IF NOT IOC$FFCHAN ()
306 1294 THEN
307 1295     BUG_CHECK (NOACPCCHAN, 'Failed to find channel for XQP');
308 1296
309 1297 CCB [CCB$B_AMOD] = -1;
310 1298
311 1299 IO_CCB = .CCB;
312 1300 IO_CHANNEL = .CHANNEL;
313 1301
314 1302 ! Initialize the rest of the impure storage area.
315 1303
316 1304
317 1305 CODE_SIZE = L_CODE_END - CODE_START;
318 1306 CODE_ADDRESS = CODE_START;
319 1307 DATA_SIZE = STORAGE_SIZE;
320 1308 DATA_ADDRESS = STORAGE_START;
321 1309
322 1310 XQP_STKLIM [0] = XQP_QUEUE;
323 1311 XQP_STKLIM [1] = XQP_STACK;
324 1312
325 1313 ! Set up the XQP queue head and dispatcher addresses.
326 1314
327 1315
328 1316 XQP_QUEUE [0] = XQP_QUEUE;
329 1317 XQP_QUEUE [1] = XQP_QUEUE;
330 1318 XQP_DISPATCHER = DISPATCH;
331 1319
332 1320 CTL$GL_F11BXQP = XQP_QUEUE;
333 1321
334 1322 END;
! end of routine INIT_STORAGE
```

.EXTRN CTL\$GL\_F11BXQP, IOC\$FFCHAN



				OBFC 00000			
57	00000000G	00	9E	00002	MOVAB	SYSEXIT, R7	
56	00000000G	00	9E	00009	MOVAB	SYSLKWSET, R6	
55		04	C2	00010	SUBL2	#4, SP	
55	F540	CA	9E	00013	MOVAB	-2752(BASE), R5	1249
53	FF4C	CA	9E	00018	MOVAB	-192(BASE), R3	
54	FF68	CA	9E	0001D	MOVAB	-152(BASE), R4	
	0000	CF	9F	00022	PUSHAB	L_CODE_START	1278
04	AE	CF	9E	00026	MOVAB	L_CODE_END-1, LOCKED_DESC+4	1279
		7E	7C	0002C	CLRG	-(SP)	1281
	08	AE	9F	0002E	PUSHAB	LOCKED_DESC	
66		03	FB	00031	CALLS	#3, SYSLKWSET	
52		50	D0	00034	MOVL	R0, STATUS	
05		52	E8	00037	BLBS	STATUS, 1\$	1282
		52	DD	0003A	PUSHL	STATUS	
67		01	FB	0003C	CALLS	#1, SYSEXIT	
6E		55	D0	0003F	MOVL	R5, LOCKED_DESC	1284
04	AE	CA	9E	00042	MOVAB	691(BASE), LOCKED_DESC+4	1285
		7E	7C	00048	CLRG	-(SP)	1287
	08	AE	9F	0004A	PUSHAB	LOCKED_DESC	
66		03	FB	0004D	CALLS	#3, SYSLKWSET	
52		50	D0	00050	MOVL	R0, STATUS	
05		52	E8	00053	BLBS	STATUS, 2\$	1288
		52	DD	00056	PUSHL	STATUS	
67		01	FB	00058	CALLS	#1, SYSEXIT	
	00000000G	00	16	0005B	JSB	IOCSFFCHAN	1293
04		50	E8	00061	BLBS	R0, 3\$	
		FEFF		00064	BUGW		1295
		0000		00066	.WORD	<BUGS_NOACPCHAN!4>	
09	A2	01	8E	00068	MNEGB	#1, 9T(CB)	1297
FF74	CA	52	D0	0006C	MOVL	CCB, -140(BASE)	1299
FF78	CA	51	D0	00071	MOVL	CHANNEL, -136(BASE)	1300
FF4C	CA	8F	D0	00076	MOVL	#<L_CODE_END-CODE_START>, -180(BASE)	1305
FF50	CA	CF	9E	0007F	MOVAB	CODE_START, -176(BASE)	1306
FF54	CA	8F	3C	00086	MOVZWL	#STORAGE_SIZE, -172(BASE)	1307
FF58	CA	55	D0	0008D	MOVL	R5, -168(BASE)	1308
	64	53	D0	00092	MOVL	R3, (R4)	1310
04	A4	55	D0	00095	MOVL	R5, 4(R4)	1311
	63	53	D0	00099	MOVL	R3, (R3)	1316
04	A3	53	D0	0009C	MOVL	R3, 4(R3)	1317
FF48	CA	CF	9E	000A0	MOVAB	DISPATCH, -184(BASE)	1318
00000000G	9F	53	D0	000A7	MOVL	R3, @NCTLSGL_F11BXQP	1320
		04	00	000AE	RET		1322

; Routine Size: 175 bytes, Routine Base: \$CODE\$ + 008D

: 335 1323 1  
: 336 1324 1 END  
: 337 1325 0 ELUDOM

## PSECT SUMMARY

Name	Bytes	Attributes
SAAAAAS	0	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(9)
\$LOCKEDC0\$	0	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(9)
\$LOCKEDC9\$	0	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$CODE\$	316	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

## Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	27	0	1000	00:01.9

## COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:INIFCP/OBJ=OBJ\$:INIFCP MSRC\$:INIFCP/UPDATE=(ENH\$:INIFCP)

Size: 300 code + 16 data bytes  
Run Time: 00:19.2  
Elapsed Time: 00:36.8  
Lines/CPU Min: 4144  
Lexemes/CPU-Min: 48519  
Memory Used: 201 pages  
Compilation Complete



0170

AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY



0171 AH-BT13A-SE  
VAX/VMS V4.0

**DIGITAL EQUIPMENT CORPORATION**  
**CONFIDENTIAL AND PROPRIETARY**